EXPRESS MAIL - MAIL LABEL NO: EL274019177US

DATE OF DEPOSIT: December 1, 2000

I hereby certify that this paper or fee is being deposited with the United States Postal Service Express Mail Post Office to Addressee service under 37 CFR §1.10 on the date indicated above and is addressed to: Box PATENT APPLICATION, Assistant Commissioner for Patents, Washington, D.C. 20231

John C. Smith, Registration No. 33,284
Printed name of person mailing paper or fee

Signature of person mailing paper or fee



INVENTORS: Peter B. Reintjes, Nicholas G. Bethmann, Shane D. Mattaway

# APPARATUS AND METHOD FOR AUTOMATIC FORM RECOGNITION AND PAGINATION

## BACKGROUND OF THE INVENTION

### Technical Field

The present invention relates to pen-based computer systems. In particular, it relates to a method and apparatus for detecting pen strokes written on a paper form by an individual and automatically determining the correct form by analyzing the pen strokes of a writer.

Background Art

There are many procedures for which the first step is an individual filling out a paper form. In most situations, the data written on the form will ultimately be input to a computer system. Historically, this has required a labor intensive two-step procedure in which a first individual fills out a form, and then a second individual enters that data into a computer database.

The development of compact high-performance computers has resulted in many new applications which heretofore were not feasible due to their demand on system resources. One such application is the pen-based, or tablet, computer. Pen-based computer applications are particularly useful in mobile environments, especially where form data is used. For example, businesses such as delivery services which previously used paper forms can now use a pen-based tablet computer to hold a paper form, and to electronically capture the data as the user fills out the form. In this type of application, the system would normally keep a copy of the form image separate from the data input by the user. This allows the input data to be more conveniently stored and/or transmitted. The development of pen-based computers has allowed data written on a form to be directly stored in a computer rather than having to take a second step to transfer the data originally written on a paper form into a computer. However, pen captured data still requires optical character recognition (OCR) software to convert the pen stroke data captured by the pen-based computer to usable data.

The next step was the development of multi-page electronic documents. This type of document may, for example, be a patient questionnaire used by physicians to obtain basic health data from a patient, or any other application which requires multiple page forms. However, as the size of a multiple page document increases, determining which

page within a document an individual is currently completing becomes more difficult. One difficulty associated with multiple page documents on tablet computers is caused by the need to identify which page in a multi-page electronic form is being filled out by an individual. It would be desirable to have a method of automatically identifying the form and individual pages so that data input by an individual could be automatically associated with the correct page of the correct form.

Another disadvantage to the prior art is that it requires explicit manual steps to initialize the data-gathering phase. For example, if a multi-part form is being used, the form must first be carefully aligned on the tablet or a registration procedure must be executed to instruct the tablet computer regarding the exact form placement. Then the specific form must be identified to the computer system. Once the form is identified, the current page must be identified whenever the user moves from one page to another in a multi-page form. As can be seen, this is a labor intensive process which is prone to error.

Unfortunately, these operations are clearly unsuitable for untrained users for the following reasons. First, there is little control over how carefully the person aligns the form. Second, we cannot ensure that an individual is taught how to identify forms or ensure that the form identification has taken place. Finally, it is not possible to ensure that an individual can be taught how to select pages or ensure that the person correctly re-selects the correct page every time they skip backwards or forwards within a multi-page form. As a result, even though electronic entry of data from forms has been developed, the process of electronically entering data written on forms has numerous disadvantages.

While addressing the basic desirability of using electronic, rather than paper forms, the prior art has failed to provide a multi-page electronic forms system which is

capable of automatically identifying a form without an individual actively taking some step or steps to identify it.

## SUMMARY OF THE INVENTION

The present invention solves the foregoing problems by providing a pen-based system that automatically identifies either a single page form or a page within a multi-page form when data is a written on paper copies of the form. The system identifies the form by identifying the area of the form on which data is written. Identification is accomplished by associating the location of pen strokes entered on the form with proper fields on the form. The system matches the location of the pen strokes input by the user with data input fields on the form that most likely match the location of the pen strokes. In one embodiment, an electronic clipboard captures pen stroke data as an untrained user fills out a form attached to its surface. The sequence and location of the raw pen-stroke data is analyzed to determine which form was filled out and which field on each page was the intended field for the subsets of stroke data. The form/field identification method allows one to use the pen-based computer as if it were an ordinary clipboard. An individual can select one of several multi-page forms, attach it onto the pen-based computer without any special attention to its positioning, fill out fields in any order, skipping between pages at will, leaving fields blank, etc. and having no other interaction with the clipboard. The resulting data is then analyzed to determine the identity of the form that was filled out by the user. The preferred embodiment uses an electronic pen that also writes normally in ink or pencil, which allows data entered by the user to be captured without requiring any additional effort from the user. In fact, the user might be unaware that the clipboard is an electronic device. Later, the collected form data can be properly superimposed on stored form template images for display.

## BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a side edge view of a prior art tablet computer which illustrates a pen-based computer that holds a multi-page form.

Figure 2 is a top view of a preferred embodiment of the invention which illustrates a pen input device filling in a text field of a form.

Figure 3 is a top view of a preferred embodiment of the invention which illustrates a boundary box calculated for a text field.

Figure 4 illustrates a form layout which has data entry locations specifically designed for use with scanning devices.

Figure 5A-B illustrates a preferred embodiment in which non-used portions of the form are blocked out to improve field recognition.

Figure 6A-B illustrates a preferred embodiment in which the standard deviation is used to define a text box.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

Prior to a discussion of the figures, a general overview of the features and advantages of the invention will be presented. The present invention relates to streamlining the process of bringing data, which is written onto forms via pen strokes,

into a computer system without requiring changes to the well-established practice of having untrained persons fill out paper forms.

Tablet computers capture data by placing a form on an electronic clipboard and then using an electronic pen that also writes normally in ink to complete the form. As the form is filled out, the data is captured without requiring any additional effort from the user. In fact, the user may even be unaware that the clipboard is an electronic device. During the initial data gathering phase, where the user is entering raw data, there are no special computational requirements. The invention provides advantages in the postprocessing of the raw data by identifying which form was filled out by the user, by associating the user entered raw data with the proper fields of the form, and by compensating for some degree of misalignment of the form on the clipboard.

The prior art approaches to solving these problems required explicit manual steps prior to allowing the user to fill out the form. For example, the form typically is carefully aligned on the tablet, or a registration procedure, such as touching the electronic pen to the upper left and lower right corners of the form would be executed to inform the tablet computer of the exact form placement. Once the placement of the form has been determined, the specific form may be identified by a variety of known techniques. Likewise, when multi-page forms are used, the current page would be identified when the user moves from one page to another. Of course, the prior art operations described above are clearly unsuitable for untrained users, because they may not properly align the form on the tablet computer, they may not know how to identify the form or how to an ensure that form identification has occurred, and they may not correctly select pages when they skip backward or forward within a multi-page form.

The principal embodiment of this invention is used in conjunction with an electronic clipboard (i.e. a tablet computer) consisting of a paper form on top of a digitizing tablet. The user fills out the form without regard to the fact that the system is also storing an electronic copy of the handwritten data. The preferred embodiment of the invention is a pen-based computer system which allows the system to identify and select a particular form based on the areas of a page or a multi-page paper document being filled out by an individual. This feature allows the user to enter data in the conventional manner on a paper form.

The present invention takes simultaneous advantage of differences between forms and differences between pages. In particular, the fact that data fields are typically located on different places on different forms allows the system to detect the correct form based on the position of the input data fields. Furthermore, the method used to do this, which is presented below, allows for the possibility that page alignment may be different for each page, or each field, if they are filled out of sequence. For example, if a person fills out a two-sided form by filling out half of the first page, flipping the page over on the clipboard and filling it out, then flipping back to the first page to complete the form, three possible page alignments must be calculated. In fact, the method described can even handle the case in which page alignment is shifted between any entries. The invention accomplishes this as follows:

Referring to Figure 1, this figure shows a side edge view of a pen-based computer 1 holding a multi-page form 3-4. The tablet computer 2 has a digitizing grid device 5 on its upper surface. In multi-page form 3-4, form 3 is folded over the end of tablet computer 2. The unused page 4 of the multi-page form 3-4 lies on top of the digitizing grid device 5. When the user fills out the unused page of multi-page form 4 with pen device 8, then the user's pen strokes are detected by the digitizing grid device 5 and their

position is used to identify what area on the form is being written on, and from that location data the system determines which form is being used.

In Figure 2, a top view of the preferred embodiment of Figure 1 is shown. The tablet computer 2 is shown securing, via optional clip 6, the filled in pages of multi-page form 3 to the digitizing grid device 5. Text input data 7 is being written by pen device 8. As the text input data 7 is written, digitizing grid device 5 detects the movement of the pen device 8 and generates an electronic representation of those movements. Digitizing grid devices 5 and their associated pen devices 8 are well-known in the art.

When the raw stroke data from the pen device 8 has been transferred to the tablet computer 2, the only information we have about this collection of data is that "someone filled out a form at a particular time." There is, however, much more information that can be obtained from the pen stroke data.

People generally fill out forms from top to bottom and from page 1 to page N in order. This behavior cannot be depended on entirely because of variations in an individual's behavior. However, it is possible to exploit this behavior without depending upon it completely. For example, address and name fields contain collections of short strokes with certain characteristics (printed letters) and numeric fields (SS#, zip codes etc.) that can be identified as such in many cases. In this case, the content of the data, as well as its location, can be used to make decisions regarding the form. If there are check boxes or circled entries (e.g. Sex M/F), they can be used as reliable indicators of which page is being used as well as the forms paper alignment on the clipboard. For the purposes of this discussion, data that can be identified based on the foregoing characteristics will the referred to as "content identifiable data."

This user input data can be used to identify the particular form and to separate the pen strokes that correspond to the fields on each particular page. Those skilled in the art will recognize that the accuracy of this approach is affected by the number of possible forms and the number of pages in the longest form. It may work very well for three to five different forms of less than four pages, while being impractical for selection between hundreds of forms with dozens of pages. However, most applications do not require a large enough number of forms that the similarity between forms would inject an element of ambiguity as to which is the correct form. Even if this were the case, it would be a simple matter to redesign the forms to avoid data field conflicts.

It is also possible to train the system in relation to a particular form. If a sample form is filled out several times in a training session, this could provide enough information for later alignment and parsing of data with a high degree of accuracy. Alternatively, a more extensive process of setting up forms by identifying fields, types of data, and the corresponding database inserts associated with post processing can make up for uncertainties in parsing the raw data. As a last resort, designing fault-tolerant form layouts may be the best approach to eliminating errors. The simplest example of this would be to have at least on required field on each form which did not overlap any other fields on the other forms.

To be fault tolerant in the case of a user who does not complete forms in a linear fashion (i.e. the user skips around between pages while filling out the form), or who leaves required fields empty, the forms should preferably be designed such that the individual field locations within the form are unique when compared with the other forms used by a particular application. Another concern is that the user may even shift the pages on the clipboard while filling out the form. In this situation, the post-processing

software may need to automatically re-register the page on a field-by-field basis when an apparent error is detected.

The process of identifying fields, and thereby identifying forms, begins by isolating blocks of writing and calculating a box around this writing for each of these blocks. Each block of writing is assigned to one of the fields in the total set of fields from all pages of all forms with the following requirements: first, no two overlapping blocks of writing can be assigned to the same field; second, all of the assigned fields must belong to the same form; and last, the linear distances between the corresponding corners of data bounding boxes and field bounding boxes should be minimized.

Processing overhead can be reduced as follows: if the text boxes and field bounding boxes perfectly match a form, then processing can terminate at that point and that form can be selected.

The boundary box 11 is illustrated in Figure 3. As can be seen, when the pen device 8 is used to enter data 7 via pen strokes on digitizing tablet 5, the system determines the perimeter borders of the text data to create a boundary box 11. Of course, the boundary box 11 is shown on this figure for illustrative purposes, but it is actually created in the host computer during post processing and does not appear on the form or the digitizing tablet.

Matching the text boundary boxes to form fields would be performed as follows. In the preferred embodiment, the search for an optimal assignment (i.e. when the field is identified) is terminated when the sum of distances between fields and data blocks falls below a predefined threshold. Further, the search will be guided by the following principles which take advantage of field placement and assume that users will generally,

but not necessarily always, fill out the form pages in the obvious order. The steps to accomplish this use the following general rules:

1- Geographical Selection Rule.
Limit initial comparisons to fields in the same region of the page as the selected data blocks (top, middle, bottom).

2- Temporal Selection Rule.
After finding a good field/block match and if the next temporal block of data is lower on the page or to the right of the previous matched block, begin by considering the next field on the same page of the same form as the field just matched. Begin alignment determination by considering zero rotation and zero X, Y offset relative to the alignment for the previous field.

3- Same Page/Next Page Rule.
More generally (than step 2, above) each good match on a page will increase the probability that the next field down the page should be matched with a field on the same page and that a following data block at the top of a page will be best matched using the page immediately following the page containing a previous good match near the bottom.

4- Same Form Rule
Each good match with a field in a given form will reduce the consideration of any fields in other forms with the exception that only a series of significantly bad matches will then allow for the consideration of alternative forms.

With these rules in mind, the process of form identification begins by isolating blocks of writing. This process is entirely different from prior art form processing systems which operate on scanned bitmaps of forms. In the prior art systems, the elements of writing must be separated from the template printed on the form and then they must be separated from each other in order to send them to a handwriting recognition system. That process is entirely different from the method implemented by the invention. In the present invention, the input data is pen stroke data representing the data entered by the user and there is no form data to be deleted.

The first part of the preferred embodiment is to isolate the segments of handwriting from each other and to identify the form and page to which these data items belong. First, the system creates an empty stroke collection which is a data structure containing a stroke list, and average character height and width and a bounding box. If this is the first stroke collection, the system will set a reasonable starting value for average character width, and then set a reasonable starting value for average character height. On the other hand, if this is not the first stroke collection, then the system takes these values from the previous stroke collection.

Next, the system sets the current position to the position of the first input stroke and adds the first stroke to the stroke collection. Subsequent strokes are added to the stroke collection until: a) The X position of the next stroke is significantly less than the X position of the previous stroke and the Y position of the next stroke is different by plus or minus 80 percent of the current average character height (e.g. this is probably an indication of a move to a new line); or b) The X position of the next stroke is significantly more than the X position of the previous stroke (e.g. probably an indication of a move to a new field on the same line).

During this process, the system also continuously recalculates average character height and width as follows:

If a new stroke S begins to the right of the end position P of all previous strokes in the current collection and no subsequent stroke begins to the left of this stroke, consider this the left hand edge of the current character. P is the right-hand edge of the previous character and S is the left-hand edge of the current character. When you have a left-hand edge and a right-hand edge of a character, use it to recompute the average character width with for example, 1/3 of the difference. Recompute the average character height by subtracting the lower bound from the upper and lower bound of all strokes in the previous collection and adding a fraction of the difference between the old character height and the new character height to the old character height.

When condition 1) or 2) above has been met:

Close the current collection and designate it as the previous collection. If the current stroke falls inside the bounding box of any previous collection, reopen that collection and make it the current collection but disable recalculation of average character height and width within this collection, otherwise, if the current stroke does not fall inside the bounding box of a previous collection, create a new stroke collection and begin adding strokes to this collection as before.

Once the collection of strokes has been closed, a box is defined which is the bounding box of all points in the collection of strokes. This bounding box may then be compared with boxes representing the fields of different forms to find the boxes that most closely match it. One embodiment of the form identification procedure finds the page for which the total distance between the corners of these bounding boxes and the field boxes

on the form template is at a minimum. The following improvement defines a different method of form identification. Since the box surrounding the handwritten text may frequently be larger than the form field because tall characters or characters with descenders, such as g or q, will frequently stray outside the bounding box of the entry field. For this reason it would be good to have a box representing the text which would not extend beyond the edges of the field even if the user's writing occasionally strays outside of that region.

The present embodiment calculates the average horizontal line above and below which the writing is formed and considers the box which extends one standard deviation above and below this average value, where the standard deviation is defined by the vertical positions of all of the other points in the stroke collection for this isolated block of writing. This is illustrated by Figures 6A-B. In Figure 6A, a portion 14 of a form is illustrated which shows the locations of data entry fields. In Figure 6B, handwritten text is shown written in the data entry fields of the portion 14 of a form. This can be seen in this figure, the lower case letter "g" has a descender which extends below its data entry field.

For a given collection of points, this embodiment calculates a box whose right and left edges are defined by the minimum and maximum X values of the data and whose top edge is defined by the average Y value of the input data plus one standard deviation of the Y values and the bottom edge is defined by the average value of Y minus the standard deviation. In the preferred embodiment, the height of the box is twice the standard deviation. However, those skilled in the art will recognize that this height can also be computed as a fraction of the standard deviation, a multiple of the standard deviation, or a different function of the variation of Y values.

The next step is to identify fields on the forms. A form is a set of scanned images, each representing one page of the form. For each page of the form, find the complete set of the largest rectangular boxes that contain no black pixels. This algorithm includes a threshold allowing it to ignore isolated islands of a few pixels that may be due to dirt on the scanner, stray marks on the form, etc.

Once the form fields are identified, the next step is to match data to the form fields. This operation matches the bounding boxes of stroke collections (as described above) with bounding boxes of fields on the forms. For a given set of raw data (e.g. a file of ink stroke data), it is assumed that all matched fields will belong to a single form.

An alternative method of associating pen stroke data with the correct form is to perform a boolean AND operation on the bitmap representing the empty form and a bitmap representing the user input data. Since the user entered data should always appear over the white spaces on the form and never (or rarely) over printing or lines on the form, the user data page which has the least number of overlapping bits with a given form page is a likely candidate for that form page. In particular, if the sum of overlapping bits for the complete set of user input pages (in order) when ANDed with the set of pages for a given form is minimum, that is the correct form.

By way of example, assume that we have three pages of writing (P1, P2, P3) with two writing segments (S1, S2) that may belong to one or more pages (S1, S2 may be on any one of the pages P1, P2, or P3). Further assume that there are 10 possible forms, of which 1-6 are three-page forms, 7-8 are two-page forms, and 9-10 are four-page forms. A comparison would be made between each page of writing with and without each of the two segments S1 and S2, to each of the corresponding pages of the three page forms. The assignment which produces the minimum number of overlapping pixels consists of

matching P1 with Form 3 Page 1, matching P2 plus the writing segment S1 matching Form 3 Page 2, and P3 plus the writing segment S2 matching Form 3 Page 3.

Assignments are only considered when all stray segments are accounted for (that is, S1 and S2 have been added to some pages and the bitmap's overlaps are still minimal), the sequence of input data pages (P1, P2, P3) is matched with the corresponding sequence of form pages (Page 1, Page 2, Page 3) and all of these pages come from the same form. For example, even if the data P2 and P3 gave zero overlapping bits when compared with Form 4 Page 2 and Form 4 Page 3, comparing P1 with the first page of Form 4 could still add enough overlapping bits so that the data/form match is not as good as another assignment.

Another important concern when implementing this system is related to page mis-alignment. This problem is minimized as follows. For every data/form page comparison, the user input bitmap should be shifted by small amounts in the X and Y directions and rotated slightly in either direction in order to find an alignment that produces the minimum number of bits overlapping. Thus each page can have a separate alignment and still match the form correctly.

In the event of ambiguity, a Page at a Time Matching with Backtracking Method is used. Because the segments of stroke data may not accurately be assigned to pages in one pass. A set of possible stroke-data to page assignments is maintained and each one is considered in turn using the Page at a Time Matching with Backtracking Method. For each assignment of ambiguous stroke data to a given page, the matches of that page are considered against stroke data that keep the minimum overlap value. If more than one assignment results in a zero overlap value these page assignment alternatives are indistinguishable and it will be necessary to flag this data set for human analysis.

This system does not assume that a person will begin writing a text segment on one page, stop writing, flip to another page and pick up writing immediately to the right of the original text segment. Therefore, this situation may cause a data set to be rejected by the automatic form identification.

Those of skilled in the art will recognize that a small set of working forms will reduce the chance of ambiguity. As the size of the working set of forms increases, the chances for ambiguity increase. As a result, when working with large sets of forms, it is important to design the forms to reduce overlap among fields in order to reduce ambiguity. Figure 4 illustrates an example of placing fields on a specific location of a form 3. The fields 12 on form 3 can be positioned so that any other form in the set of possible forms will not have long text fields in the same location as on this form, thereby reducing form ambiguity.

The electronic capture of form data and its combination with an electronic form is preferably transparent to the user. If pen device 8 is an electronic pen that also writes in ink on the paper form, data can be captured when the user is filling out a paper form without requiring any additional effort from the user. In fact, the user might even be unaware that the tablet computer 2 is an electronic device. For example, as an untrained user fills out a paper form attached to its surface, digitizing device 5 captures pen stroke data without the user's knowledge. The sequence and location of the raw pen-stroke data is analyzed to determine which form was filled out and which field on each page was the intended field for the subsets of stroke data. The form/field identification method allows one to use the tablet computer 2 as if it were an ordinary clipboard. An individual can select one of several multi-page forms, attach it onto the clipboard without any special attention to its positioning, fill out fields in any order, skipping between pages at will, leaving fields blank, etc. and having no other interaction with the tablet computer 2.

In Figure 5A-B, another preferred embodiment is illustrated which improves field recognition. Form field identification can be improved by matching the input data against specially modified versions of the form in which all of the forbidden regions (where user input should not appear) are completely blacked out. In Figure 5A, the form 3 is shown which has boxes 12 designed to hold raw data 13 (handwriting entries). To improve recognition of forbidden areas, Figure 5B illustrates a form 3 in which the forbidden areas are blacked out. By using this template, the bitmap AND comparison will produce many more overlapping pixels and more easily identify inappropriate data field assignments. The result of ANDing these templates with the average data regions will still produce few or zero overlapping pixels for the correct form, but much greater numbers of overlapping pixels for incorrect forms.

While the invention has been described with respect to a preferred embodiment thereof, it will be understood by those skilled in the art that various changes in detail may be made therein without departing from the spirit, scope, and teaching of the invention. For example, the software platform may be anything suitable for pen-based computers, the type of output device used to indicate status can vary, etc. Accordingly, the invention herein disclosed is to be limited only as specified in the following claims.

We claim: